

ALCF I/O Library

Bill Allcock – LCF

ESP April Workshop: "Code for Q"

May 2, 2012

Purpose

- Real-time collection of IO events
 - Possible coordination of IO
 - Debug of IO subsystem
- Improve analysis of IO subsystem
 - Help correlate with Darshan data
 - Understand what “type” of IO users perform
- This is a work in progress, so we are still investigating what and how to make this library as useful as possible to LCF and to LCF users.



Benefits

- Currently these are all forward looking ideas and do not exist today
- Coordination with scheduler
 - If the scheduler knows that a job is currently in an IO phase, can defer termination of job.
- Coordination with “other” IO
 - LCF system IO operations can be deferred/suspended when applications are in IO

Concept and Implementation

- Concept
 - Users bracket their major IO routines with “start” and “end” calls.
 - These routines might also do computation or communication
 - We definitely don’t want to bracket individual IO operations
- Implementation
 - C, C++ and FORTRAN
 - No requirement for MPI



Integration

- General
 - Copy 'alcf.h' (for C) and 'alcf-f90.h' (for FORTRAN) into your source tree
 - Add 'alcf_init' to beginning of the application
 - Add 'alcf_finalize' to the end of the application
 - Instrument your IO phases with 'alcf_io_start' and 'alcf_io_end'
- Site Specific
 - Add '-DALCF' to compiler options to turn on ALCF library
 - Add library and path to linker options.
 - -L/soft/apps/libalcf/lib
 - -lalcf



Example C

```
#include "alcf.h"
int main (int argc, char **argv)
{
    int id;
    char filename[80];

    alcf_init();

    alcf_io_start(ALCF_IO_CHECKPOINT, ALCF_IO_WRITE,
filename, "this is an important file", &id);
    do_checkpoint();
    alcf_io_end(id);

    alcf_finalize();

    return 0;
}
```

Example - FORTRAN

```
program test
  #include <alcf-f90.h>

  INTEGER ID
  CHARACTER          FNAME*20
  CHARACTER          COMMENT*20

  FNAME = 'foo.txt'
  COMMENT = 'my big checkpoint'

  CALL_ALCF_INIT()

  CALL_ALCF_IO_START(ALCF_IO_LOG, ALCF_IO_WRITE, FNAME, COMMENT,
ID)
  CALL_ALCF_IO_END(ID)

  CALL_ALCF_FINALIZE()

  stop
end
```

Documentation

- `alcf_init`
 - Call once per process to initialize library and resources
- `alcf_finalize`
 - Call once per process to release resources
- `alcf_io_start`
 - Call before starting major IO phase
 - Returns a transaction Id that is used in corresponding `_end()`
 - Four input arguments
 - Type – one of our pre-defined types, (checkpoint, analysis, etc)
 - Direction – read and/or write
 - Filename – name of file
 - Comment – any interesting information about this IO phase.
- `alcf_io_end`
 - Call at end of major IO phase
 - Uses transaction Id from `_start()`

